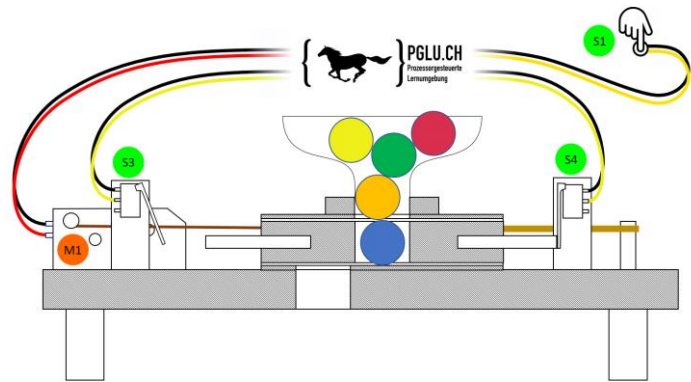


# 1

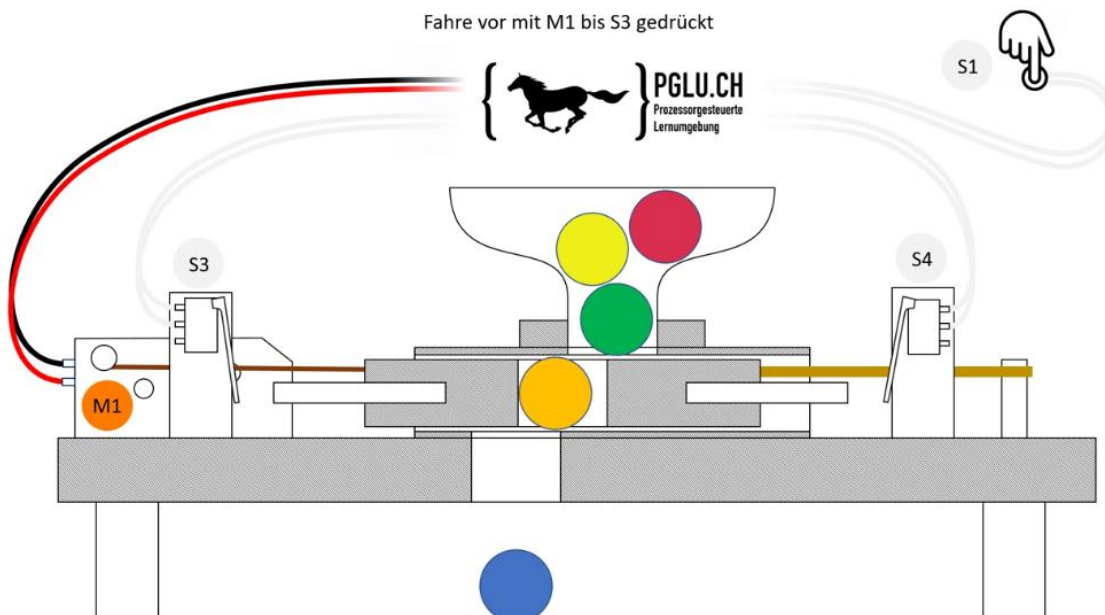


**Aufgabe** Beschreibe den Kaugummiautomaten und das was er tut!

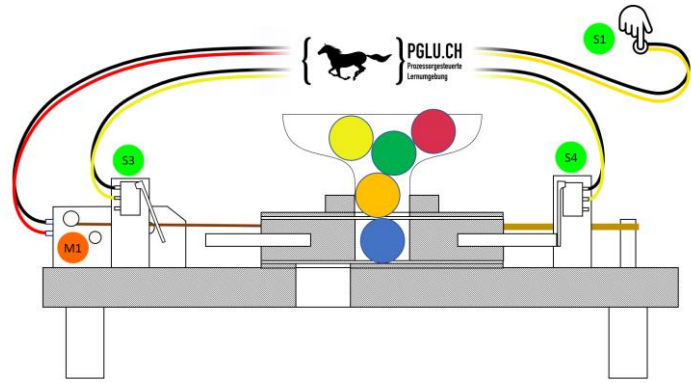
Schaue die Animation und beobachte was geschieht. Stelle dir vor, du müsstest den Kaugummiautomaten einer Person am Telefon erklären, um ihn ihr zu verkaufen.

Schreibe einen Text dazu!

- Aus welchen Einzelteilen besteht der Kaugummiautomat?
- Welche Wege werden gefahren?
- Wann fällt ein Kaugummi in den Schieber?
- Wann fällt ein Kaugummi aus dem Schieber heraus?



# 2



**Aufgabe** Schreibe das Programm, indem du es in drei Sequenzen aufteilst. Diese kannst du auch in der Animation beobachten. Setze sie nacheinander in den Hauptloop ein!

1. Warte bis Taster an S1 gedrückt wird. Fahre dann vor
2. Fahre vor bis Endschalter S3 erreicht ist. Ein Kaugummi fällt raus. Stoppe dann
3. Fahre rück bis Endschalter S4 erreicht ist. Ein neuer Kaugummi fällt hinein. Stoppe dann

Teste dein Programm im Simulator und drücke nacheinander: S1, S3, S4. Die Werte von M1 müssen dann so sein: 0%, 100% -70%, 0%

Während dem Hochladen des Programms auf die Platine: Schalte den Motor immer mit dem linken Kippschalter aus!

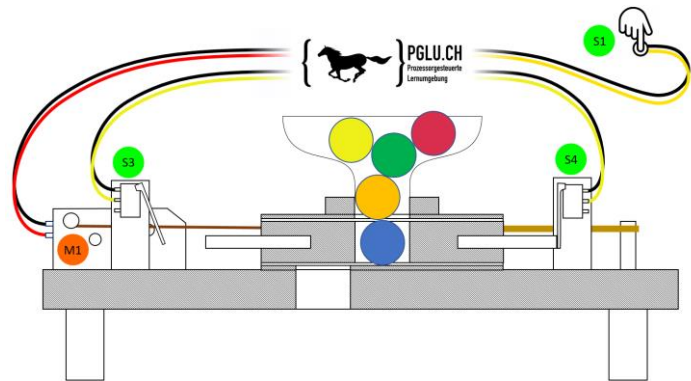
The screenshot shows the PGLU.CH programming interface with the following components:

- Navigation:** Programmieren (Grafik, Code), Testen (Simulator), Organisieren (Programme), Hilfe (?).
- Settings:** Hauptloop: 100'000 mal pro Sekunde, Blinkcode: kurz 1, lang 1.
- Sequenz 1:**
  - setze Motor 1 auf 100 %
  - wiederhole bis: setze Motor 1 auf 0 %
  - Sensor 1 = EIN prüfe entprellt 50ms
- Sequenz 2:**
  - wiederhole bis: setze Motor 1 auf 100 %
  - Sensor 3 = EIN prüfe entprellt 50ms
  - setze Motor 1 auf 0 %
- Sequenz 3:**
  - wiederhole bis: setze Motor 1 auf -70 %
  - Sensor 4 = EIN prüfe entprellt 50ms
  - setze Motor 1 auf 0 %

Annotations in the image:

- Yellow boxes highlight the three sequences.
- Text: "Warum nur -70% beim Rückfahren?" points to the -70% value in Sequenz 3.
- Text: "Weil der Gummi mithilft!" points to the -70% value in Sequenz 3.

# 3



**Aufgabe** Mache dein Programm übersichtlicher und eleganter

- Nutze für jede der 3 Sequenzen eine eigene Funktion!
- Hole dazu im Menu «Funktion» zuerst eine Klammer und gebe ihr einen Namen.
- Ist die Funktion benannt, findest du im Menu einen neuen Block, den du in den Hauptloop einsetzen kannst!

Während dem Hochladen des Programms auf die Platine: Schalte den Motor immer mit dem linken Kippschalter aus!

The screenshot shows the PGLU.CH programming environment. The main loop is set to '100'000 mal pro Sekunde' with blinkcode 'kurz 1' and 'lang 1'. The main loop contains three function blocks: 'Stoppe bis S1 gedrückt', 'Fahre vor bis S3 erreicht', and 'Fahre rück bis S4 erreicht'. The left sidebar shows the 'Funktion' menu. The main workspace displays three sequences:

- Sequenz 1:** A 'wiederhole bis' loop containing 'setze Motor 1 auf 0 %' and 'Sensor 1 = EIN prüfe entprellt 50ms', followed by 'setze Motor 1 auf 100 %'.
- Sequenz 2:** A 'wiederhole bis' loop containing 'setze Motor 1 auf 0 %' and 'Sensor 3 = EIN prüfe entprellt 50ms', followed by 'setze Motor 1 auf 100 %'.
- Sequenz 3:** 'Sensor 4 = EIN prüfe entprellt 50ms', followed by a 'wiederhole bis' loop containing 'setze Motor 1 auf -70 %', and finally 'setze Motor 1 auf 0 %'.

Yellow lines connect the function blocks in the main loop to their corresponding sequence blocks.